



# **Introduction to MVIS ROS DRIVER nodes**

Version 1.2  
Apr 2025

**MicroVision, Inc.**  
**18390 NE 68th St**  
**Redmond, WA 98052**  
**U.S.A.**  
**Phone: +1 (425)-936-6847**  
**E-mail: [info@microvision.com](mailto:info@microvision.com)**

Copyright © 2025 MicroVision, Inc., Redmond, U.S.A. (hereinafter "MicroVision"). All rights reserved.

NOTICE: All information contained herein is, and remains the property of MicroVision and its suppliers, if any. The intellectual and technical concepts contained herein are proprietary to MicroVision and its suppliers and may be covered by the U.S., German and foreign patents, patents in process, and are protected by patent or copyright laws. This User Manual contains trade secrets and protected information that is MicroVision privileged or confidential information and all contents shall not be disclosed to any third party. Dissemination of this information or reproduction of this material is strictly forbidden unless prior written permission is obtained from MicroVision, except as allowed under copyright laws.

The information contained herein is accurate and reliable to the best of our knowledge at the time of printing and is subject to change without notice. No responsibility is assumed by MicroVision for use of this information. MicroVision, its logo and MOVIA are trademarks of MicroVision. All other trademarks are the property of their respective companies. Specifications subject to change without notice.

public

## Version History

Date	Version	Changes
Feb-06-2025	1.0	Initial Version of this manual
Apr-29-2025	1.2	Movia L Image; ROS path corrections

Table 0.1: Version History

## Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Prerequisites</b>	<b>6</b>
<b>3</b>	<b>ROS Driver</b>	<b>7</b>
3.1	Build . . . . .	7
3.2	ROS . . . . .	7
3.2.1	RViz visualisation . . . . .	8
3.3	ROS2 . . . . .	9
3.3.1	RViz2 visualisation . . . . .	9
3.4	Dependencies . . . . .	10
3.4.1	Required SDK plugins . . . . .	10
3.4.2	Build SDK sources . . . . .	10
3.5	Configure <b>movia</b> driver node . . . . .	11
3.5.1	Setup movia node configuration parameters . . . . .	11
<b>A</b>	<b>Troubleshooting</b>	<b>12</b>
A.1	Retry build . . . . .	12
A.2	Windows . . . . .	12
A.2.1	Target Platform . . . . .	12

## 1 Introduction

This document explains the general usage of the MVIS ROS DRIVER nodes and how they can be adapted.

The MVIS ROS driver nodes are available for ROS and ROS2.

Please read the appendix if you experience problems.

## 2 Prerequisites

Before building the ROS and ROS2 driver nodes ROS has to be installed and working on your system. Try building one of the ROS tutorials first.

### 3 ROS Driver

This MOVIA driver package contains ROS and ROS2 **movia** nodes to integrate a MOVIA-sensor into an existing ROS or ROS2 environment.

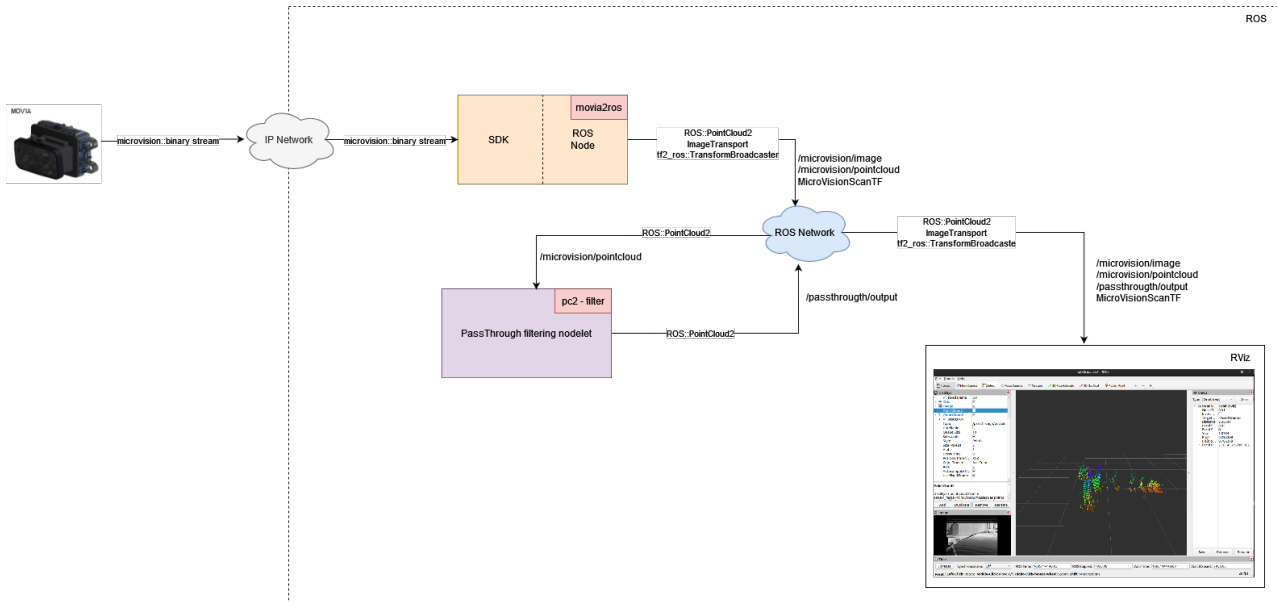


Figure 3.1: movia\_driver package integration

Binary stream of a connected MOVIA device is converted by SDK into a public known datatype which is converted by this movia driver node in a second step into ROS standard datatype PointCloud2. For MOVIA L only UDP is used and Scan2342 is received from the sensor instead of raw LDML. Also intensity images are received.

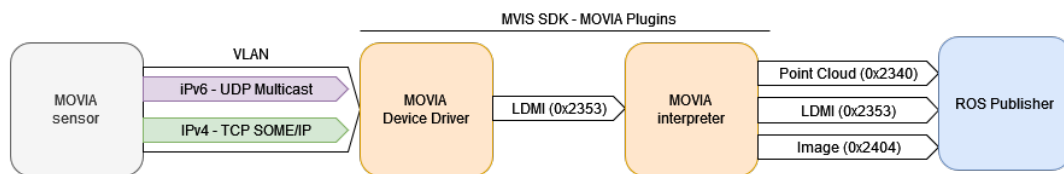


Figure 3.2: movia driver data flow

#### 3.1 Build

With the build script you can build the node for your ROS version and platform. Only if using a not supported platform you have to build the ros nodes manually.

#### 3.2 ROS

The **MOVIA** driver requires ROS and the ROS PCL installed. To learn how to build and launch ROS nodes please also refer to these tutorials: <http://wiki.ros.org/ROS/Tutorials> There is also an optional pcl2 filter nodelet, please

refer to [http://wiki.ros.org/pcl\\_ros/](http://wiki.ros.org/pcl_ros/)

### 3.2.1 RViz visualisation

To visualize the PointCloud the RViz ROS visualization tool can be used. To start RViz in the second terminal window please proceed as follows (example for noetic):

```
source /opt/ros/noetic/setup.bash  
rviz&
```

movia\_driver package also contains prepared RViz configuration:

```
{YOUR_SDK_PATH}/mvis-movia-driver/src/movia/config/movia.rviz
```



Example of RViz output:

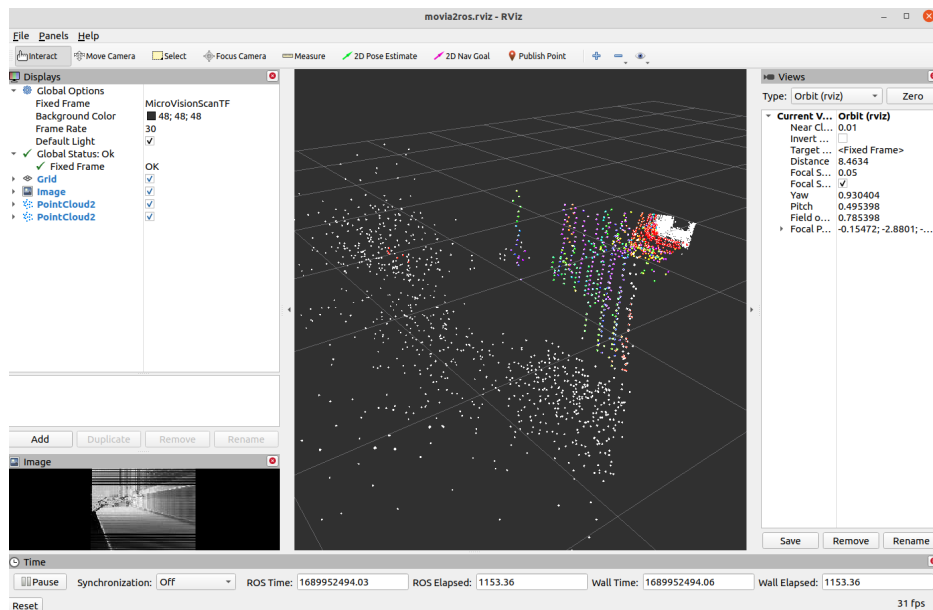


Figure 3.3: movia and RViz visualisation

### 3.3 ROS2

The **MOVIA** driver requires ROS2 installed. To learn how to build and launch ROS2 nodes please also refer to these tutorials for each distribution. As an example, here is a tutorial for Humble Hawksbill distribution: <https://docs.ros.org/en/humble/Tutorials.html>

#### 3.3.1 RViz2 visualisation

To visualize the PointCloud the RViz2 ROS2 visualization tool can be used. To start RViz2 in the second terminal window please proceed as follows (example for humble):

```
source /opt/ros/humble/setup.bash
rviz2&
```

movia package also contains a prepared RViz2 configuration:

```
ros/src/movia/launch/movia.rviz
ros2/rviz_config/default.rviz
```

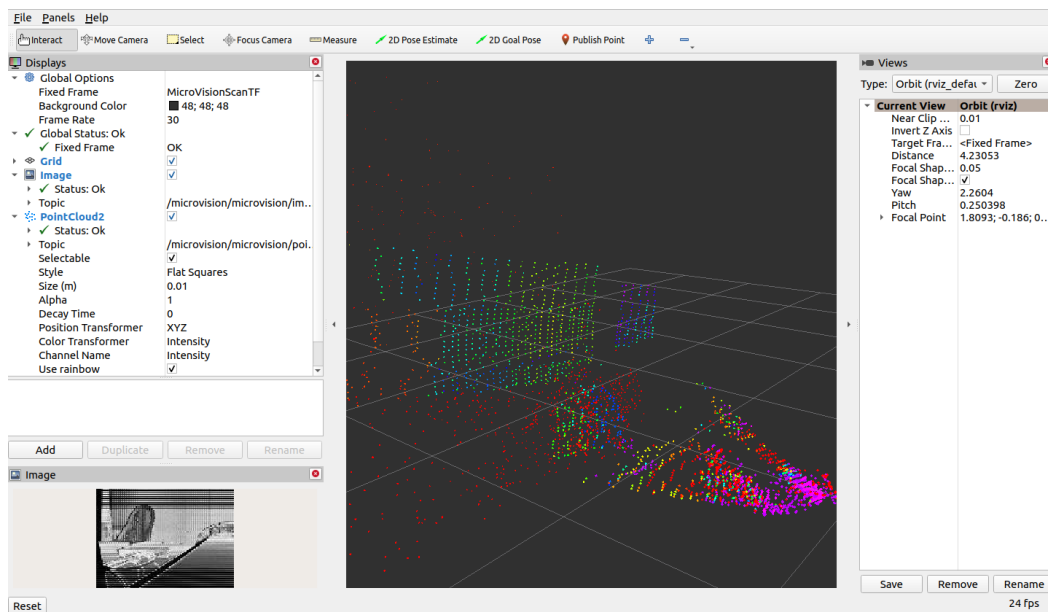


Figure 3.4: movia and RViz2 visualisation

### 3.4 Dependencies

While the mvis-ros-driver package comes with all the required binary dependencies for the supported platforms you may have to collect them for yourself when using other platforms. You may be unable to use the build scripts.

#### 3.4.1 Required SDK plugins

For MOVIA sensor support, several plugins need to be loaded.



#### CAUTION

To use the MicroVision MOVIA Lidar sensor, the following plugins are required:

- *movia\_interpreter-plugin*
- *movia\_device-plugin* and
- *thirdparty\_networkinterface-plugin*. (optional)

These plugins are contained in the MVIS SDK plugins download file which is available on the website separately.

Please have a look inside the plugin user manuals for more details!

#### 3.4.2 Build SDK sources

Prerequisite for building the **movia\_driver** packages is the built SDK version. Please build SDK sources as "release" configuration:

```
./build.sh -c Release
```

Please note the referred build.sh is the one in the MVIS C++ SDK.

### 3.5 Configure movia driver node

For most use cases you can simply use the run script with or without arguments.

#### 3.5.1 Setup movia node configuration parameters

If you need to modify more parameters than available in the run script you can modify the movia.yaml configuration in install folder and adjust the values like scope\_id, fov, etc.:

```
# MOVIA sensor hardware id (HWID printed on casing or "L" for MOVIA L)
hwid: "11" # or "L" for MOVIA L

# multicast ip address of sensor (with scope id if not MOVIA L)
multicast_ip: "ff02::1be0:1%8"

# remote ip address of sensor depending on vlan setup (not required for MOVIA L)
remote_ip: "172.16.101.56"

# Optional: Set to false to use local time of host otherwise sensor time is used
sensor_timesynced: false
```

## A Troubleshooting

Please set the default log level to 'Debug' and read the messages carefully. Most likely some required library is not found!

**CMake log** Add `-trace` to `build.bat` `catkin` or `colcon cmake-args` to get more information about the build process. Also you can add `-DCMAKE_MESSAGE_LOG_LEVEL=VERBOSE` to get the full build output.

### A.1 Retry build

Please delete the folders `build`, `install`, `log` in the `ros` and `ros2` subdirectories before running build again. This is required when switching `ros` distribution or changing other build settings.

### A.2 Windows

**Requirements** Boost and `libjpeg-turbo` may need to be installed on your system to build the nodes. See MVIS SDK documentation.

**Debugging** You can find information on debugging ROS nodes in the ROS documentation. We tested debugging with Visual Studio Code. Use a 'RelWithDebInfo' target platform.

#### A.2.1 Target Platform

It is important to use the same target platform for all libraries and your node. Please build the SDK for 'Release-shared' to avoid crashes in the ROS node. Use `RelWithDebInfo` for debugging. Make sure the MVIS SDK is built with the same settings!

#### **cmake-args**

When building your ROS or ROS2 node with 'catkin\_make install' or 'colcon build' the target platform can be set by adding the `cmake-args`:

```
--cmake-args -DCMAKE_BUILD_TYPE=RelWithDebInfo
```

For more log output (check the `CMAKE_PREFIX_PATH`!) during the build adjust the log-level:

```
--log-level=VERBOSE
```